APPENDIX XII - RESOURCE DEPLETION EVENT


```
import java.lang.*;
import java.util.*;

/**
 * Event returned by the system when an uncaught exception or error is encountered.
 */
public class ResourceDepletionEvent implements IMessageEvent
{
   public final static int RESOURCE_SYS_MEM_DEPLETED = 0X40000001;
   public final static int RESOURCE_CPU_BANDWIDTH_DEPLETED = 0X40000002;
   public final static int RESOURCE_OUT_OF_SYS_MEM = 0X40000100;

   // Static error event Id number.  Incremented for each new event.
   private static long lastEventId;

   private long eventId;
   private long appId;
   private int typeCode;
   private long date;
   private String message;

   /**
    * Create a SystemErrorEvent.
    *
    * @param errorType - The value indicating the error; see IMessageEvent.
    *
    * @param errorMsg - The readable error message in the following format:
    */
   public ResourceDepletionEvent(int typeCodeIn, long appIdIn, String messageIn)
   {
      lastEventId = (lastEventId == Long.MAX_VALUE) ? 0 : lastEventId + 1;
      eventId = lastEventId;
      typeCode = typeCodeIn;
      date = System.currentTimeMillis();
      message = messageIn;
   }

   /**
    * Get the unique identifier of the event.
    *
    * @return unique identifier of the event.
    */
   public long getEventId()
   {
      return eventId;
   }

   /**
    * Get the unique identifier of the application that logged the event.
    *
    * @return The application identifier.
    */
   public long getAppId()
```

```
    {
      return appId;
    }

    /**
     * Get the event type code.  Identifies a code specific to the event system.
     *
     * @return type code of the event.
     */
    public int getTypeCode()
    {
      return typeCode;
    }

    /**
     * Set the error type code.  Allows an application to change the error received.
     *
     * @param error - The new error value.
     */
    public void setTypeCode(int typeCodeIn)
    {
      typeCode = typeCodeIn;
    }

    /**
     * Get the event date in milli-seconds from midnight January 1, 1970.
     *
     * @return Date the event was submitted to the system.
     */
    public long getDate()
    {
      return date;
    }

    /**
     * Set the date.  Allows an application to change the date received.
     *
     * @param error - The new error value.
     */
    public void setDate(long dateIn)
    {
      date = dateIn;
    }

    /**
     * Get the readable message.
     *
     * @return the readable event message.
     */
    public String getMessage()
    {
      return message;
    }

    /**
     * Set the error message string.  Allows an application to change the error message received.
     *
```

```
     * @param error - The new error message.
     */
    public void setErrorMessage(String messageIn)
    {
      message = messageIn;
    }

    protected void finalize()
    {
      // Store the lastErrorEventId in a persistent storage object.
    }

    static {
      if(/*some persistent storage object exists with error event Id stored in it*/ true)
        lastEventId = /* value in the persistent storage object */1;
      else
        lastEventId = 0;
    }
}
```